

Bit Bucket x'3F'

Ed Jaffe, edjaffe@phoenixsoftware.com
Art Gutowski, Arthur.Gutowski@huntington.com
Mike Shorkend, mike.shorkend@gmail.com
James Lund, james-lund@tamu.edu



SHARE 138
Session 393
Dallas, TX
30 Mar 2022





UNIX Back-Up Catch-Up (Ed Jaffe)

z/OS UNIX Backup Current Status

- We were early adopters of HSM z/OS UNIX file backup support
- I reported much difficulty attempting to use this support in my z/OS 2.4 User Experience presentation at SHARE 2 ½ years ago
- There have been many APARs taken since that time and IMHO we are finally starting to see the light at the end of the tunnel
- Two APARs are must-haves before I can recommend its use:
 - OA61080 - Slow z/OS UNIX backup performance (OPEN, no PTFs yet)
 - OA61711 - HRECOVERed files are corrupt (HIPER - PTFs Available)
 - Get current with everything else HSM related! (Lotsa OC4s & such)
- I also recommend splitting your existing HSM into CLASSIC Mode and FILE Mode HSMs (FILE Mode HSM handles z/OS UNIX only)
 - Won't impact your existing HSM workload
 - Each mode has unique MCDS, BCDS, OCDS & JRNL



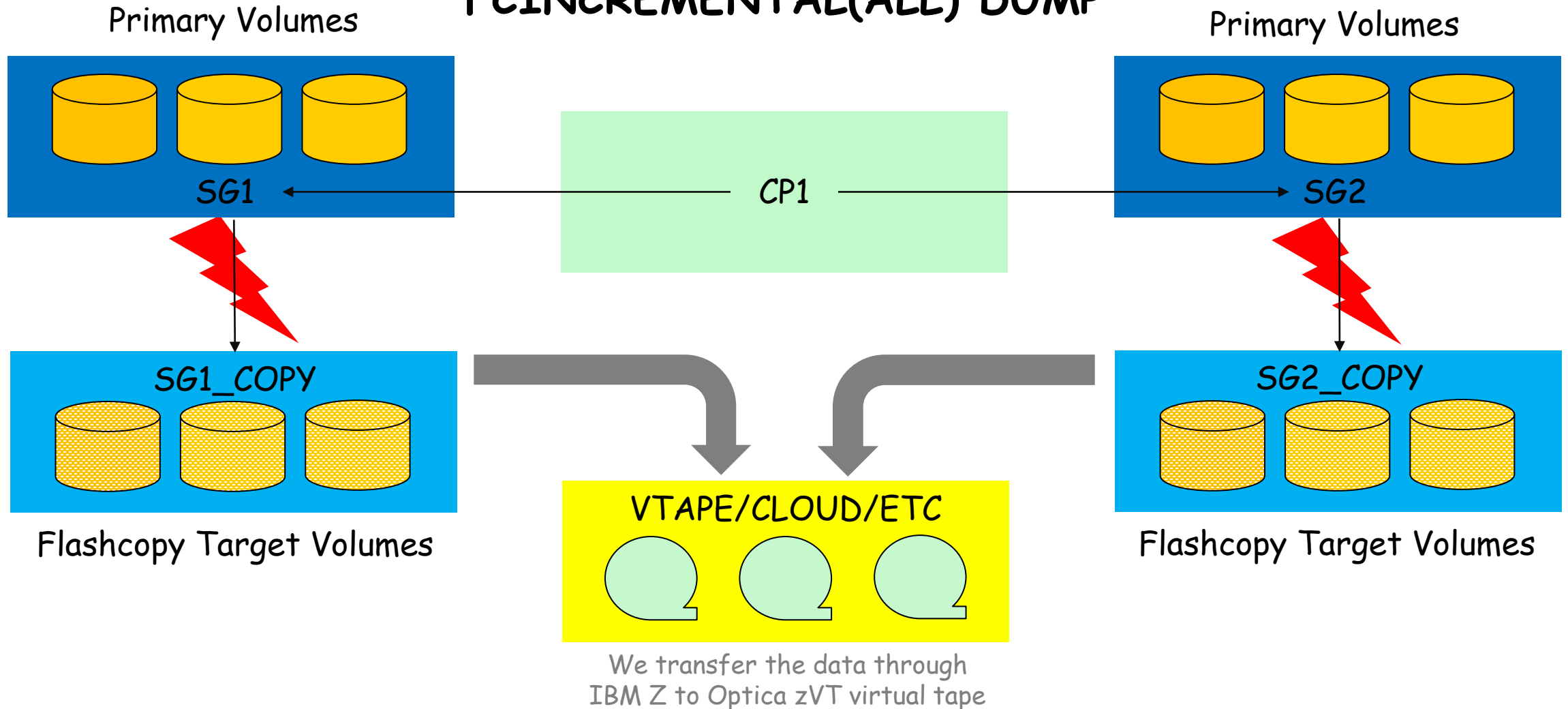
Done in a Flash
(Ed Jaffe)

Fast-Replicating HSM Full Volume Dumps For Non-SMS Volumes

- For the past several years, HSM development has been steadily adding and improving support for fast replication full-volume dump of SMS-managed volumes
- This support relies upon SMS constructs (storage groups, copy pools, and copy pool storage groups) as well as a suite of "FR" commands (FRBACKUP, FRDELETE, FRRECOV)
- No similar support exists for Non-SMS managed volumes
 - However, there is concurrent copy support e.g.,
SETSYS VOLUMEDUMP(ANYPREFERRED)
- We wanted a fast-replication solution for our Non-SMS managed z/OS volumes that was still (mostly) under HSM control so we have reporting, automatic dump volume expiration, command-based volume restore, etc.

FRBACKUP (SMS Volume) Overview

FRBACKUP CP(CP1) EXECUTE
FCINCREMENTAL(ALL) DUMP



What We Did To Simulate This for Non-SMS Volumes

- We defined Flashcopy target volumes (one for each source volume)
 - Descriptive volume names were chosen to make them easy to recognize on an HSM report since HSM has no awareness of this relationship
- We added the Flashcopy target volumes to HSM as primary volumes eligible for automatic dump and no other management activities
- We created DSS JCL (started via Brian Westerman's AUTO), that runs a few moments before HSM automatic dump executes
- This JCL initiates incremental Flashcopy operations as needed
 - We specify DUMPCOND(itioning) on the COPY commands to ensure no CLIP (i.e., DSF REFORMAT) needed if volume must be restored

PARALLEL

```
COPY ALLD(*) ALLX ADMIN IDY(A4RES1) ODY(#A4RS1) PRG FCINCR DUMPCOND
COPY ALLD(*) ALLX ADMIN IDY(A4RES2) ODY(#A4RS2) PRG FCINCR DUMPCOND
COPY ALLD(*) ALLX ADMIN IDY(A4SYS1) ODY(#A4SYS) PRG FCINCR DUMPCOND
... etc.
```

Restoring z/OS Volumes Dumped Using This Process

- If the volume can be restored with the system up, use the standard HSM RECOVER command with TOVOLUME(flashcopyvolser) and TARGETVOLUME(originalvolser)
 - E.g., TOVOLUME(#A4RS1) TARGETVOLUME(A4RES1)
- If you must use DSS JCL to restore the volume, specifying COPYVOLID will restore the *source* volser and not the Flashcopy target volser

```
//INPUT DD DSN=DFHSM.DMP.dumpclass.V#A4RS1.Dyyddd.Thhmmss,  
...  
... (etc)  
...  
//SYSIN DD *  
  RESTORE FULL INDDNAME(INPUT) OUTDYNAM(MYVOL) ADMIN PURGE CPYV  
//
```

 This will be volume A4RES1 after restore

Dealing With z/VSE and Linux for Z Volumes

- z/VSE and Linux for Z volumes are handled similarly
- Neither have indexed VTOCs, so you cannot use DUMPCOND

PARALLEL

COPY ALLD(*) ALLX ADMIN IDY(DOS620) ODY(#OS620) PRG FCINCR

COPY ALLD(*) ALLX ADMIN IDY(LNX400) ODY(#NX400) PRG FCINCR

... etc.

- Expect a warning for the z/VSE volumes (no Format-5 Label)

0ADR310W (022)-DDTFP(01), ERROR FOUND IN VTOC. UNALLOCATED SPACE WILL BE PROCESSED FOR VOLUME DOS620

- Backup is performed using the HSM BACKVOL command

BACKVOL DUMP(DUMPCCLASS(dclass1)) UNIT(3390) -
VOLUMES(#OS620 #NX400 etc...)

- CLIP required after HSM RECOVER command (due to implied CPYV)
- No CLIP required after DSS JCL RESTORE if OUTDYNAM references the correct volser and COPYVOLID not specified

Dealing With z/VM Volumes

- Many z/VM volumes have no VTOC so we copy them by TRACKS

```
COPY TRKS(0,0,10016,14) ADMIN CPVOL IDY(710RES) ODY(#710RS) PRG FCINCR  
COPY TRKS(0,0,10016,14) ADMIN CPVOL IDY(710COM) ODY(#710CM) PRG FCINCR
```

- We dump them up to large, zEDC-compressed, SMS-managed DASD data sets and immediately MIGRATE them using an HSM command

```
DUMP TRKS(0,0,10016,14) IDY(#710RS) ODD(DASD) ADMIN CPVOL OPT(4) ZCOMP(PREF)  
...  
HSEND WAIT MIGRATE DSN(&DSPFX..VMFVD.D&DATEQ..V&V) ML2
```

- We explicitly re-MIGRATE the DASD backup after a DSS restore

```
RESTORE TRKS(0,0,10016,14) IDD(DASD) ODY(&VOLSER.) PRG ADMIN CPVOL  
...  
HSEND WAIT MIGRATE DSN(&DSPFX..VMFVD.D&DATEQ..V&V) ML2
```

Summary



- It is possible to use HSM to manage fast-replication full volume dumps of Non-SMS managed z/OS volumes
- You need your own job to initiate the Flashcopy each day
- The only “strangeness” is the appearance of the Flashcopy target volume names in reports and in RECOVER/RESTORE commands
- Extend this concept to cover z/VSE and Linux for Z volumes
- z/VM volumes are strange “beasts,” but still managed easily
- Bottom Line: You can obtain fast-replication point-in-time FVDs of your entire mainframe “world” from z/OS using HSM (and probably its ISV competitor products as well)

Acknowledgements Knowing and Unknowing

- Glenn Wilcock (IBM)
- Max Smith (IBM)
- Andrew Wilt (IBM)
- Amy Gilbert (Phoenix Software - Retired)



Check Your Syntax and
Datasets at the Door...

(Art Gutowski)

Attributions, Trademarks, Service Marks



- IBM®, MVS™, z/OS®, IBM Z®, IBM z Systems®, and ISPF® are registered trademarks or trademarks of International Business Machines Corporation registered in many jurisdictions worldwide.
- All other trademarks, service marks, and company names are properties of their respective owners.

ISPF is DEAD?



Long live ISPF!

IBM's Little-known Symbolic PARMLIB Parser

- Included as-is with ISPF (not officially supported)
- Installs in a jiff
- Checks your Symbol table based on LOADPARM specification, and can perform symbolic substitutions and run the entire LOADxx->IEASYMxx->IEASYSxx-> chain
- "But Mulder, you'll say..." what about existing "check" parms on SET commands?
- "But Scully, look at the evidence..."
 - there are only 3 so far (IEFOPZ, PROG, and SMFLIM),
 - they didn't exist when SPP was written
 - they *only* syntax check
 - SPP can perform additional validations - range/value, PARMLIB members and datasets
- "But Mulder, you'll say..." isn't SETOMVS SYNTAXCHECK thorough enough?
- "Yes, Scully," I'll say, "that's true... that is one shining exception, however..."
 - it still only allows you to check based on CURRENT symbol values - you may have to update your Symbol Table first... sometimes that's OK, sometimes it's not
 - SPP allows you to specify a "staged" Symbol table via an alternate LOADxx to see what the entire IPL would look like...
- So, lets get on with it...

SPP Installation...

- The installation EXEC and package information is supplied in `SYS1.SAMPLIB`, and creates the execution libraries under your `USERID` (or `TSO PREFIX`) as an `HLQ`
- Use `/R` to refresh after an upgrade

```
Menu Options View Utilities Compilers Help
DSLIST - Data Sets Matched in list SPPPARM          Some entries not used
Command ==> _____ Scroll ==> CSR

Total Tracks:      2163 non-x:      2163 Data Sets:      3 non-x:      3
-----
Command - Enter "/" to select action              Message          Volume
          Tracks %      XT Device  Dsorg Recfm Lrecl Blksz  Created    Referred
-----
ex /(sppinst) '''sys1.samplib(spppack)'''          SYXALX
          1863 76      1 3390      PO  FB          80 27920 2022/03/03 2022/03/21
-----
          SYS5.SPP.PRD.PARMLIB                      Z5SY02
          150 30      1 3390      PO  FB          80 27920 2022/03/17 2022/03/18
-----
          SYS5.SPP.STG.PARMLIB                      Z5SY02
          150 1      1 3390      PO  FB          80 27920 2022/03/17 2022/03/18
***** End of Data Set list *****

Software Package Installation Exec V0.30 (c)1990,1995 IBM Corp.

Allocating Dataset: 'SYS5.PARMLIB.MESSAGES'
Allocating Dataset: 'SYS5.PARMLIB.PANELS'
***
```

```
Creating: 'SYS5.PARMLIB.NOTES(INSTALL)'
Creating: 'SYS5.PARMLIB.NOTES(INSTRUCT)'
Creating: 'SYS5.PARMLIB.NOTES(LIMITS)'
Creating: 'SYS5.PARMLIB.NOTES(WORKFLOW)'

SYS1.PARMLIB Processor Instructions

o The installation tool has allocated the datasets necessary
  to execute the SYS1.PARMLIB Processor Dialog.

o To use the Tool, you will need to execute the following
  command from ISPF option 6 or the equivalent.

  COMMAND ==> EX 'myid.PARMLIB.EXEC(SYSPARM)'
  or
  COMMAND ==> EX 'myid.myfiles.EXEC(SYSPARM)' 'myid.myfiles'

o These instructions and additional information regarding the
  tool have been copied into the PDS: PARMLIB.NOTES. All members
  in this dataset should be read prior to executing the tool.

***
```

- Yada, yada, 5 screens of messages later, (should've used `/Q`), it's ready to use, almost...

SPP Setup - Test PARMLIB Concatenation

```
Menu Options View Utilities Compilers Help
DSLIST - Data Sets Matched in list SPPPARM Row 1 of 7
Command ==> Scroll ==> CSR

Total Tracks:      2429 non-x:      2429 Data Sets:      7 non-x:      7
-----
Command - Enter "/" to select action Message Volume
Tracks % XT Device Dsorg Recfm Lrecl Blksz Created Referred
-----
SYS1.SAMPLIB SYXALX
1863 76 1 3390 PO FB 80 27920 2022/03/03 2022/03/21
-----
SYS5.PARMLIB.EXEC Z5SY01
148 10 1 3390 PO VB 255 3360 2022/03/16 2022/03/21
-----
SYS5.PARMLIB.MESSAGES Z5SY06
54 1 1 3390 PO FB 80 3120 2022/03/16 2022/03/21
-----
SYS5.PARMLIB.NOTES Z5SY06
2 50 1 3390 PO VB 255 23476 2022/03/16 2022/03/21
-----
SYS5.PARMLIB.PANELS Z5SY07
62 11 1 3390 PO FB 80 3120 2022/03/16 2022/03/21
-----
SYS5.SPP.PRD.PARMLIB Z5SY02
150 30 1 3390 PO FB 80 27920 2022/03/17 2022/03/18
-----
SYS5.SPP.STG.PARMLIB Z5SY02
150 1 1 3390 PO FB 80 27920 2022/03/17 2022/03/18
-----
***** End of Data Set list *****
*WORK
```

- Recommended reading:
- PARMLIB.NOTES(*)
- it's not that long
- Also documented in
z/OS MVS
Initialization &
Tuning Reference
(SA23-1380),
Appendix B

SPP Setup - LOADxx

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW      SYS5.SPP.PRD.PARMLIB(LOADAG) - 01.00      Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000010 *****
000020 * PARMLIB(LOADAG):
000030 *   This is a demonstration-only LOAD Parm for use with IBM's
000040 *   Symbolic Parmlib Parser
000050 *
000060 * Modification Log:
000070 *   yyyy-dd-mm   Who   What
000080 *   -----
000090 *   2022-03-18   AJG   Created from Production LOAD Parm
000091 *****
000092 * LOAD Parameters for ALL Images
000093 *****
000100 ARCHLVL  2
000200 NUCLEUS  1
000300 NUCLST   01
000310 *****
000700 * LPAR-Specific LOAD Parameters
001000 * !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
001100 LPARNAME TESTLP1
001200 IODF     == SYS1      TESTPLEX 00 Y 0
001300 PARMLIB  SYS5.SPP.STG.PARMLIB
001310 PARMLIB  SYS5.SPP.PRD.PARMLIB
001400 PARMLIB  SYS1.IBM.PARMLIB      *****
001500 SYSCAT   SYXMCT133CCAT.ICF.TESTPLEX.MASTER      CAT
001600 IEASYM    (AG,L)
001800 SYSPLEX  TESTPLEX
*WORK
```

- In PARMLIB concatenation, rather than SYSn.IPLPARM
- see PARMLIB.NOTES, member LIMITS

SPP Usage - Light it up

```
Menu Options View Utilities Compilers Help
-----
DSLIS - Data Sets Matched in list SPPARM Row 1 of 7
Command ==> _____ Scroll ==> CSR

Total Tracks:      2429 non-x:      2429 Data Sets:      7 non-x:      7
-----
Command - Enter "/" to select action Message Volume
Tracks % XT Device Dsorg Recfm Lrecl Blksz Created Referred
-----
SYS1.SAMPLIB SYXALX
1863 76 1 3390 PO FB 80 27920 2022/03/03 2022/03/21
-----
ex / (sysparm) 'sys5.parmlib' Z5SY01
148 10 1 3390 PO VB 255 3360 2022/03/16 2022/03/21
-----
SYS5.PARMLIB.MESSAGES Z5SY06
54 1 1 3390 PO FB 80 3120 2022/03/16 2022/03/21
-----
SYS5.PARMLIB.NOTES Z5SY06
2 50 1 3390 PO VB 255 23476 2022/03/16 2022/03/21
-----
SYS5.PARMLIB.PANELS Z5SY07
62 11 1 3390 PO FB 80 3120 2022/03/16 2022/03/21
-----
SYS5.SPP.PRD.PARMLIB Z5SY02
150 60 1 3390 PO FB 80 27920 2022/03/17 2022/03/21
-----
SYS5.SPP.STG.PARMLIB Z5SY02
150 2 1 3390 PO FB 80 27920 2022/03/17 2022/03/21
-----
***** End of Data Set list *****
*WORK
```

- The positional parm "Name" is required once I set my TSO PREFIX back to my USERID

SPP Usage - Member Selection (Primary) Panel

```
----- Parmlib Processor Member Selection Panel -----
COMMAND ==>

Member Name          ==> ?          [Enter '?' for supported member list]
Browse, Edit or View ==>           [Optional. Default is VIEW for symbol
                                   substitution, and EDIT otherwise.]

Dsn ==> 'sys5.spp.prd.parmlib'      Dflt: 'SYS1.PARMLIB'
Volume of above Dsn  ==>           [Optional]
SYS1.NUCLEUS Volume  ==>           [If different from one catalogued]
FMID                 ==>           [Optional. Enter '?' for list]

Do Symbol Substitution ==> y        [Y or N]
Symbol Substitution Values:
                                   [Use '*' for Current values below]
LOADxx Member        ==> loadag     Default LOAD00. LOADA1 <=== Current
Hardware Name        ==> proc01     [Optional]          PROC03 <=== Current
LPAR Name            ==> testlp1    [Optional]          <=== Current
VM Userid            ==>           [Optional]          <=== Current
SYSRES Volume        ==>           [Optional]          SYXALX <=== Current
Master Cat Volume    ==>           [Optional - Substituted for *MCAT* ]

Help is available on all Panels via PF1 or HELP on the command line.
```

*WORK

- SPP “discovers” and sets some defaults
- It keeps track in ISPF Profile member IPXPROF
- Point it where you want to start, or ? for a member selection list
- I let the mode default
- To invoke symbol substitution, also specify LOADXX, HW name, and LPAR name...

SPP Usage - Wow, this thing is good

- Found a ghost in the machine, right out of the gate...

```
ARC1020I DFSMSHSM IS RECALLING FROM TAPE DSN=SYS3.IPLPARM, YOU MAY CONTINUE THE RECALL IN  
THE BACKGROUND AND FREE YOUR TSO SESSION BY PRESSING THE ATTENTION KEY  
ARC1010I USER REQUEST FOR A MIGRATED DATA SET FAILED.  
ARC1001I SYS3.IPLPARM RECALL FAILED, RC=0002, REAS=0000  
ARC1102I DATA SET IS NOT MIGRATED/BACKED UP  
IEA013E ERROR DETECTED PROCESSING IEASYMXX  
008 DATA FOR SYSPARM IN ERROR:  
      (AG,&LPN.,L)  
006 DATA LENGTH ERROR FOR SYMBOL: &BMCLVL.  
***
```

- Not a showstopper for SPP, but have something needing attention...
- Then, the real (staged) errors...

SPP Usage - Fix IEASYM first

```
----- Parmlib Processor Member Selection      IEASYMxx Errors
COMMAND ==>

Member Name      ==> IEASYMAG  (Enter '?' for supported member list)
Browse, Edit or View ==> edit   (Optional. Default is VIEW for symbol
                                substitution, and EDIT otherwise.)

Dsn ==> 'SYS5.SPP.STG.PARMLIB'      Dflt: 'SYS1.PARMLIB'
Volume of above Dsn ==>              (Optional)
SYS1.NUCLEUS Volume ==>              (If different from one catalogued)
FMID ==>                            (Optional. Enter '?' for list)

Do Symbol Substitution ==> n        (Y or N)
Symbol Substitution Values:
                                (Use '*' for Current values below)
LOADxx Member      ==> LOADAG  Default LOAD00.  LOADA1  <=== Current
Hardware Name      ==> PROC01  (Optional)      PROC03  <=== Current
LPAR Name          ==> TESTLP1 (Optional)      <=== Current
VM Userid          ==>         (Optional)      <=== Current
SYSRES Volume      ==>         (Optional)      SYXALX  <=== Current
Master Cat Volume  ==>         (Optional - Substituted for *MCAT* )

Help is available on all Panels via PF1 or HELP on the command line.
```

Error Messages were issued during IEASYMxx processing. Correct the errors and rerun the substitution.

*WORK

- So, set the Member Name...
- My IEASYM is in the STG PARMLIB
- Set Edit mode
- Set Symbolic Substitution to "N" for this fix
- NOTE: Checking IEASYM with Symbol Substitution ON is not terribly productive...
 - Use SYMSHOW / IPXSHOW command instead

SPP Usage - Fixing IEASYM...

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      SYS5.SPP.STG.PARMLIB(IEASYMAG) - 01.02      Columns 00001 00072
Command ==> Scroll ==> CSR
000100 SYSDEF                                     /* DC1 - PROD */
000200 HNAME(PROC01)
- - - - - 6 Line(s) not Displayed
000900 SYMDEF(&GDPSCFGA.='NORM')                 /* GDPS ACF/CFG FILE SUFFIX */
001000
001100 SYSDEF                                     /* DC3 - DR */
001200 HNAME(PROC02)
- - - - - 6 Line(s) not Displayed
001900 SYMDEF(&GDPSCFGA.='ALT')                   /* GDPS ACF/CFG FILE SUFFIX */
002000
002100 SYSDEF                                     /* DC1 - PROD */
002200 HNAME(PROC03)
- - - - - 7 Line(s) not Displayed
003000 SYMDEF(&GDPSCFGA.='NORM')                 /* GDPS ACF/CFG FILE SUFFIX */
003100
- - - - - 2 Line(s) not Displayed
003400 SYSDEF
003500 SYSPARM(AG,&LPN.,L)
003600 SYMDEF(&PRIHSM.='N')
- - - - - 1 Line(s) not Displayed
003800 SYMDEF(&LPN.='&SYSNAME(-1:1).')
- - - - - 95 Line(s) not Displayed
013500 SYSDEF
013600 LPARNAME(TESTLP1)
013700 SYSNAME(TLP1)
013800 SYMDEF(&BMCLVL.='V060300C')                 /* BMC */
014000 SYMDEF(&MQLVLT.='PA')                     /* MQSERIES */
*WORK
```

- Using the error messages as a guide
 - fix up SYSPARM (can't use symbolic here)
 - fix the &BMCLVL symbol (text too long)

SPP Usage - Fixing IEASYM...

- "End" forces me to save or cancel my updates... nice...

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      SYS5.SPP.STG.PARMLIB(IEASYMAG) - 01.04      Data changed-SAVE/CANCEL
Command ==>  Scroll ==> CSR
***** ***** Top of Data *****
- - - - - 12 Line(s) not Displayed
000100 SYSDEF      /* DC1 - PROD */
000200 HNAME( PROC01)
- - - - - 6 Line(s) not Displayed
000900 SYMDEF(&GDPSCFGA,='NORM') /* GDPS ACF/CFG FILE SUFFIX */
- - - - - 1 Line(s) not Displayed
001100 SYSDEF      /* DC3 - DR */
001200 HNAME( PROC02)
- - - - - 6 Line(s) not Displayed
001900 SYMDEF(&GDPSCFGA,='ALT') /* GDPS ACF/CFG FILE SUFFIX */
- - - - - 1 Line(s) not Displayed
002100 SYSDEF      /* DC1 - PROD */
002200 HNAME( PROC03)
- - - - - 7 Line(s) not Displayed
003000 SYMDEF(&GDPSCFGA,='NORM') /* GDPS ACF/CFG FILE SUFFIX */
- - - - - 3 Line(s) not Displayed
003400 SYSDEF
003500 SYSPARM(AG,L)
- - - - - 2 Line(s) not Displayed
003800 SYMDEF(&LPN,='&SYSNAME(-1:1).')
- - - - - 95 Line(s) not Displayed
013500 SYSDEF
- - - - - 1 Line(s) not Displayed
013700 SYSNAME(TLP1)
013800 SYMDEF(&BMCLVL,='C') /* BMC */
- - - - - 14 Line(s) not Displayed
*WORK
```

SPP Usage - "Restart"...

```
----- Parmlib Processor Member Selection Panel -----
COMMAND ==>

Member Name      ==> ?      (Enter '?' for supported member list)
Browse, Edit or View ==> edit (Optional. Default is VIEW for symbol
                               substitution, and EDIT otherwise.)

Dsn ==> 'SYS5.SPP.PRD.PARMLIB' Dflt: 'SYS1.PARMLIB'
Volume of above Dsn ==>      (Optional)
SYS1.NUCLEUS Volume ==>      (If different from one catalogued)
FMID             ==>      (Optional. Enter '?' for list)

Do Symbol Substitution ==> Y (Y or N)
Symbol Substitution Values:
                               (Use '*' for Current values below)
LOADxx Member      ==> LOADAG Default LOAD00. LOADA1 <=== Current
Hardware Name      ==> PROC01 (Optional)      PROC03 <=== Current
LPAR Name          ==> TESTLP1 (Optional)      <=== Current
VM Userid          ==>      (Optional)      <=== Current
SYSRES Volume      ==>      (Optional)      SYXALX <=== Current
Master Cat Volume  ==>      (Optional - Substituted for *MCAT* )

Help is available on all Panels via PF1 or HELP on the command line.

*WORK      -SDSF
```

- After fixing and saving IEASYM, it's back to member selection
- DO the symbolic substitution
- And the PRD PARMLIB
- But this time, I'll Edit...

SPP Usage - On to IEASYS

```
----- PARMLIB Processor ----- Row 1 of 66
Supported Member List Panel

COMMAND ==>                                SCROLL ==> HALF

Select a Parmlib Type with any character

Option  Member      System Parameter  Description ('*' for only symbol substitution)
-----  -
LOADxx   IPL Parameters
IEASYMxx  System Symbols
S  IEASYSxx SYSP=      * - System Parameters
ALLOCxx  ALLOC=      * - Allocation Parameters
AUTORxx  AUTOR=      * - AUTOR Parameters
AXRxx    AXR=       * - AXR Parameters
BPXPRMxx OMVS=      * - Open MVS Parameters
CEAPRMxx CEA=       * - CEA Parameters
CEEPRMxx CEE=       * - CEE Parameters
CLOCKxx  CLOCK=     * - Clock Parameters
COMMNDxx CMD=       * - System Commands
CONSOLxx CON=       * - Console Parameters
COUPLExx COUPLE=    * - Coupling Parameters
CSVRTLxx RTLS=      * - RTLS Parameters
DIAGxx   DIAG=      * - Diagnostic Parameters
EXITxx   EXIT=      * - Dynamic Exit Parameters
FXEPRMxx FXE=       * - FXE Parameters
GRSCNFxx GRSCNF=    * - GRS Configuration
GRSRNLxx GRSRNL=    * - GRS RNL Parameters
GTZPRMxx GTZ=       * - Generalized Tracker

*WORK
```

- Now, we can walk through each member individually, or
- Start with IEASYS and run the chain...

SPP Usage - IEASYS

```
----- PARMLIB Processor ----- Row 1 of 4
Suffix Selection Panel

COMMAND ==>                                SCROLL ==> HALF
```

```
      Select a Parmlib Member with B, E/S, or V (V for Symbol Substitution)
      Primary Parmlib Dataset = SYS5.SPP.PRD.PARMLIB
      Primary Parmlib VOLUME =
      Member Type             = IEASYSxx
      NEW IEASYSxx SUFFIX ==>
```

A	Name	VV	MM	Created	Changed	Size	Init	Mod	ID
S	IEASYSAG	01.35	16/03/28	22/03/18	13:24	28	24	28	H020628
	IEASYSDR	01.03	19/04/30	21/01/22	10:12	1	1	1	HB31943
	IEASYSX0	01.33	16/03/28	22/03/07	10:27	16	24	0	H003654
	IEASYS00	01.62	88/04/04	22/01/27	15:52	56	36	31	H020628

```
***** Bottom of data *****
```

- and pick my test
SYS parms

SPP Usage - IEASYS - so far, so good...

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW      SYS5.SPP.STG.PARMLIB(IEASYSAG) - 01.36      Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
=NOTE= The following Line was originally:
=NOTE=  CMD={&LPN.C,0V,8J},
000100  CMD={1C,0V,8J},
000200  COUPLE=AG,
000300  DEVSUP=00,
=NOTE= The following Line was originally:
=NOTE=  DIAG={00,&LPN.2},
000400  DIAG={00,12},
000500  GRSRNL=00,
000600  PROD=A0,
000700  PROG={A0,ZA,CT},
000800  OMVS={F0,S0},
=NOTE= The following Line was originally:
=NOTE=  PAGE={PAGE.&SYSNAME..PLPA,
000900  PAGE={PAGE.TLP1.PLPA,
=NOTE= The following Line was originally:
=NOTE=  PAGE.&SYSNAME..COMMON,
001000  PAGE.TLP1.COMMON,
=NOTE= The following Line was originally:

IEASYSAG being processed in data set 'SYS5.SPP.STG.PARMLIB' on Volume
Z5SY02. Issue SYMSUB with the cursor at the beginning of a supported
parameter to process that member(s). Issue DOALL to process all members
pointed to by this member.

=NOTE=  PAGE.&SYSNAME..LOCAL3,L),
*WORK
```

- I get a warning message en route that IEASYSAG is not in the concatenation for this IPL, but that's intentional
- Note that I'm in View mode (I didn't specify a new member suffix)
- I can SYMSUB to check individual members
- For time's sake, let's DOALL and see what happens...

SPP Usage - DOALL

```
----- PARMLIB Processor -----  
Symbol Substitution Error Panel  
  
COMMAND ==>  
  
An ERROR was detected while performing Symbolic Substitution:  
  
Derived Member 'COMMND1C' could not be found anywhere in the  
Parmlib Concatenation.  
  
  
Press the HELP Key [PF1] to view Help for Symbolic Substitution  
Press the END Key [PF3] or Enter to continue
```

*WORK

- That doesn't look good...
- I'll fix that later (using Edit an No Symbol Substitution)
- On to the next...

SPP Usage - Running the IEASYS Chain

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW      SYS5.SPP.PRD.PARMLIB(COMMNDQV) - 01.04                      No Errors
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -CAUTION- Profile changed to NUMBER OFF (from NUMBER ON STD).
==MSG>      Data does not have valid standard numbers.
000001 COM='V C321-C325,OFFLINE'
000002 COM='V C328,OFFLINE'
000003 COM='V C22D,OFFLINE'
000004 COM='V C31F,OFFLINE'
000005 COM='V C414,OFFLINE'
000006 COM='V C514,OFFLINE'
000007 COM='V C614,OFFLINE'
000008 COM='V C714,OFFLINE'
***** ***** Bottom of Data *****
```

*WORK

- Not much to see here... good...
- Let's skip ahead to something interesting

SPP Usage - SYMCHECK, aka IPXCHECK

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW      SYS5.SPP.STG.PARMLIB(COUPLEAG) - 01.48      No Errors
Command ==> ipxcheck      Scroll ==> CSR
***** Top of Data *****
000100 /*-----*/
000200 /* PARMLIB(COUPPEAG): */
000300 /* This is a demonstration-only Sysplex COUPLE parameter set for */
000400 /* use with IBMs Symbolic PARMLIB Parser */
000500 /* */
000600 /* Modification Log: */
000610 /* yyyy-mm-dd Who What */
000620 /* ----- */
000630 /* 2022-03-18 AJG Created */
000640 /* */
000650 /*-----*/
000660 /* */
=NOTE= The following Line was originally:
=NOTE= COUPLE SYSPLEX(&SYSPLEX.)
000800 COUPLE SYSPLEX(TESTPLEX)
000900 CFRMPOL(CFRMX6)
=NOTE= The following Line was originally:
=NOTE= PCOUPLE(SYS1,XCF,&SYSPLEX.,PRIM,CDS)
001000 PCOUPLE(SYS1,XCF,TESTPLEX,PRIM,CDS)
=NOTE= The following Line was originally:
=NOTE= ACOUPLE(SYS1,XCF,&SYSPLEX.,ALT,CDS)
001100 ACOUPLE(SYS1,XCF,TESTPLEX,ALT,CDS)
001200 OPNOTIFY(180)
001300 MAXMSG(8000)
001400 INTERVAL(165)
001500 FUNCTIONS ENABLE(CRITICALPAGING)
*WORK
```

- COUPLE member symbols validated OK, but lets' make sure my datasets are there
- SYMCHECK is a macro that parses a string for a dataset name, and checks to see if it exists
- So, enter the command, place the cursor on the parm and...

SPP Usage - IPXCHECK

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW      SYS5.SPP.STG.PARMLIB(COUPLEAG) - 01.48      Dataset NOT found
Command ==> Scroll ==> CSR
***** Top of Data *****
000100 /*-----*/
000200 /* PARMLIB(COUPLEAG): */
000300 /* This is a demonstration-only Sysplex COUPLE parameter set for */
000400 /* use with IBMs Symbolic PARMLIB Parser */
000500 /* */
000600 /* Modification Log: */
000610 /* yyyy-mm-dd Who What */
000620 /* ----- */
000630 /* 2022-03-18 AJG Created */
000640 /* */
000650 /*-----*/
000660 /* */
=NOTE= The following Line was originally:
=NOTE= COUPLE SYSPLEX(&SYSPLEX.)
000800 COUPLE SYSPLEX(TESTPLEX)
000900 CFRMPOL(CFRMX6)
=NOTE= The following Line was originally:
=NOTE= PCOUPLE(SYS1.XCF.&SYSPLEX.,PRIM.CDS)
001000 PCOUPLE(SYS1.XCF.TESTPLEX.PRIM.CDS)
=NOTE= The following Line was originally:
=NOTE= ACOUPLE(SYS1.XCF.&SYSPLEX.,ALT.CDS)
001100 ACOUPLE(SYS1.XCF.TESTPLEX.ALT.CDS)
001200 OPNOTIFY(180)
001300 MAXMSG(8000)
001400 INTERVAL(165)
001500 FUNCTIONS ENABLE(CRITICALPAGING)
*WORK
```

- Oh, Lucy, looks like I have some allocating to do...
- This works great here, and PROGxx, but...

SPP Usage - IPXCHECK

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW      SYS5.SPP.STG.PARMLIB(BPXPRMS0) - 02.11      Columns 00001 00072
Command ==> ipxcheck      Scroll ==> CSR
007600
007700
=NOTE= The following Line was originally:
=NOTE=  ROOT  FILESYSTEM('SYS1.OMVS.&SYSPLEX..ROOT')
007800  ROOT  FILESYSTEM('SYS1.OMVS.TESTPLEX.ROOT')
007900          TYPE(ZFS)
008000          MODE(READ)
008100
=NOTE= The following Line was originally:
=NOTE=  MOUNT FILESYSTEM('SYS1.OMVS.&SYSNAME..SYSTEM.ROOT')
008200  MOUNT FILESYSTEM('SYS1.OMVS.TLP1.SYSTEM.ROOT')
=NOTE= The following Line was originally:
=NOTE=  MOUNTPOINT('/&SYSNAME.')
008300  MOUNTPOINT('/TLP1')
008400  TYPE(ZFS)
008500  MODE(RDWR)
008600  UNMOUNT
008700
```

- Not so well for BPXPRMxx
- The required quote marks for the FILESYSTEM parm throw it off

```
IKJ56709I INVALID DATA SET NAME, 'SYS1.OMVS.TESTPLEX.ROOT'
***
```

SPP Usage - The GRS RNL checker

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW      SYS1.IBM.PARMLIB[GRSRNL00] - 01.00      No Errors
Command ==> _____ Scroll ==> CSR
***** Top of Data *****
==MSG> Parsing was successful.
000100 /*****
000200 /*  START OF SPECIFICATIONS                               */
000300 /*
000400 /*01  MACRO-NAME = GRSRNL00                               */
000500 /*
000600 /*01  MACRO-DATE = 07/11/08                               @P2C*/
000700 /*
000800 /*01  DESCRIPTIVE-NAME = GRS RESOURCE NAME LIST MACRO     */
000900 /*
000910 /*01  COPYRIGHT =                                         */
000920 /*    LICENSED MATERIALS - PROPERTY OF IBM               */
000930 /*    5694-A01 COPYRIGHT IBM CORP. 1980, 2009            */
000940 /*
000950 /*01  STATUS = HBB7760                                     @P2C*/
000960 /*
000970 /*01  FUNCTION = THE GRSRNL00 CONTAINS THE DEFAULT CONTENTS FOR
000980 /*                      GRSRNLXX MEMBER OF SYS1.PARMLIB (I.E., THE SYSTEM
000990 /*                      INCLUSION, THE SYSTEMS EXCLUSION, AND THE RESERVE
001000 /*                      CONVERSION RESOURCE NAME LISTS).   */
001100 /*
001200 /*
001300 /*01  MODULE-TYPE = MACRO                                   */
001400 /*
001500 /*
001600 /*01  PROCESSOR = ASSEMBLER                                */
*WORK
```

- SPP Provides an RNL parsing tool, ISGPARMC, which validates the definitions, not just the symbol substitutions

SPP Usage - Missing symbols

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW      SYS5.SPP.PRD.PARMLIB(PROGA0) - 01.27      Problems Found
Command ==> _____ Scroll ==> CSR
***** Top of Data *****
==MSG> -CAUTION- Profile changed to NUMBER ON STD (from NUMBER OFF).
==MSG> Data has valid standard numbers.
000100 /* */
000200 /* */
000300 /* */
000400 /* */
000500 /* */
000510 /* */
000600 /* */
000700 /* APF LIST DEFINITIONS */
000800 /* */
000900 APF FORMAT(DYNAMIC)
=NOTE= The following Line was originally:
=NOTE= APF ADD DSNAME(CGU.ZMF.E.USERCUST.LINKLIB.&ZMFTST.) SMS
001110 APF ADD DSNAME(CGU.ZMF.E.USERCUST.LINKLIB.V825) SMS
=NOTE= The following Line was originally:
=NOTE= APF ADD DSNAME(CGU.ZMF.E.USERVEND.LINKLIB.&ZMFTST.) SMS
001120 APF ADD DSNAME(CGU.ZMF.E.USERVEND.LINKLIB.V825) SMS
==MSG> WARNING: Symbol NOT substituted.
001130 APF ADD DSNAME(CGP.ZMF.E.PRODCUST.LINKLIB.&ZMFPRD.) SMS
==MSG> WARNING: Symbol NOT substituted.
001140 APF ADD DSNAME(CGP.ZMF.E.PRODVEND.LINKLIB.&ZMFPRD.) SMS
=NOTE= The follow
=NOTE= APF ADD DS 4 Warning(s) and 0 Error(s) detected. MS
001400 APF ADD DS
=NOTE= The following Line was originally:
*WORK
```

- SPP's bread and butter
- Forgot to define a release
- Add this to the list of fixes
- Again, SYMCHECK / IPXCHECK can be used here to validate resolved datasets

SPP Usage - Miscellaneous



- I didn't show it here, but the LOADxx editor is neat, but use with caution
 - It doesn't handle multiple filters (HWNAME, LPARNAME)
 - In EDIT mode, if you don't CANCEL, it will overwrite your current LOADxx unless you specify a new suffix
 - It doesn't preserve YOUR comments
 - It affords you an edit session to pre-pend comments, but
 - It creates its' own comments between statements
 - If you select View mode and No for symbol substitution, it places you in an ISPF View session

SPP Usage - Additional Commands

Symbol Substitution Commands:

- SYMSUB** - Place the cursor under a supported SYSP Parm and it will display the member(s) represented by the Parm with symbol substitution.
- SYMSHOW** - Shows the current symbols being used for substitution.
- SYMWRITE {DSN}** - Writes the current symbol table in binary format to DSN if supplied or uses the default of 'prefix.SYMBOL.TABLE'.
- SYMCHECK** - Verifies that the dataset at the cursor position is valid on the current system. Accepts most PARMLIB dataset syntax including optional Volume specifications.
- IPXDOSUB** - Performs Symbol Substitution on any dataset, using a previously created symbol table.
- DOALL {ERRORS}** - Starting from the View Panel, this command will examine all the datasets that an IEASYSxx member points to. If ERRORS is coded, only members with problems will be displayed. Otherwise all members will be displayed.

SPP Usage - And an API

Symbol Substitution Commands (Continued):

IPXMMRET - Programming Interface to place a copy of the Symbol Table built by the Tool into Virtual Storage. Can be used as input to **ASASYMBM** for programs that wish to 'roll their own' substitution for other processes. Must be called in an ISPF Environment. No input is required. Return Code 0 in General Register 15 indicates success. Returns a structure pointed to by General Register 1. First word has Subpool and Length of structure. Subsequent words are the Symbol Table. Structure Storage must be freed by the user (Storage is task related, so it will be freed automatically at task termination).

The Interface is fully described in Callable Services.

SPP - Want to see more?

- Lobby IBM to make it a supported tool
 - Write a SHARE RFE - ask me or Mike Shorkend how
 - <https://www.share.org/Connect/Advocacy-Requirements-System>
- CUCI, CUCI, lend me your mod...
 - Tom Conley's ISPF Cabal is actively writing ISPF enhancements
 - See <https://cbttape.org/ftp/cbt/CBT967.zip>
- Lobby IBM to open it up
 - The OpenMainframe project - build a repository and Git-r-done
- RYO
 - If you're familiar with Rexx and ISPF Dialog Manager, and you want to learn about the aforementioned IPXMMRET, you can do it!

Acknowledgements Knowing and Unknowing

- Tom Conley
- Mike Shorkend
- Chris Carter

Workload Interaction Correlator

(Mike Shorkend)

What is WIC?



Marna WALLE (mwalle) @mwalle · Dec 23, 2021

...

If you are an RMF customer, here's a great new function that you can try out, at no cost to you.



Daniel Rosa @danielvrosa · Dec 23, 2021

IBM has announced changes to the IBM z/OS Workload Interaction Correlator offering so clients running z/OS V2.4 with a RMF license or z/OS V2.5 with a RMF or an Advanced Data Gatherer (ADG) license are entitled to the Correlator at no additional charge.

Recommendations for z/OS SMF 98 Records



March 29, 2021



Community ▾

Search

IBM Z and LinuxONE Community

Participate ▾

Topic groups ▾

User groups

New Enhancements Delivered for z/OS Workload Interaction Correlator and Navigator

By [Daniel Wiegand](#) posted Mon November 29, 2021 12:44 PM

2

Recommend

What is SMF 98?

- WIC is a system component that records SMF record type 98
- Db2, CICS and IMS write their own 98 records (more subsystems to come?)
- These records can show the correlation between different system events which can help us find the root cause of operational issues
- To view these events you need a product like IBM z/OS Workload Interaction Navigator(licensed)
- IBM recommends that you collect 98s even if you do not have Navigator. It can help IBM with trouble shooting

WIC pre-reqs

- You need to be licensed for RMF 2.4 or higher
 - You have to be in SMF logstream mode
 - You need an in-memory resource for WIC records
 - SMFPRMxx needs the WIC Keyword(default is NOWIC)
 - SET SMF=xx
- Or
- SETSMF WIC(requires AUTHSETSMF)
 - Update IFAPRDxx and enable WIC (only if you are licensed for RMF 2.4 or higher)

Getting WIC running(1/3)

Before

```
D SMF,WIC
IFA714I 13.47.09 SMF STATUS 435
SMF WIC STATUS
  SPECIFIED SMF PARAMETER: NOWIC
  WorkloadIntCorr PRODUCT FEATURE: DISABLED
ST   K PG # AS  ROUTINE  VERSION  R A E LAST ROUTINE CALL T
-----
00001          IEAHFXSV          N      0000/00/00 00:00:00
```

Getting WIC running(2/3)

PARMLIB(IFAPRDxx) (only required parameters shown)

```
EDIT          SYSx.PLXx.PARMLIB(IFAPRD00) - 01.06          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
000532  PRODUCT OWNER('IBM CORP')
000533      NAME('z/OS')
000534      ID(5650-Z0S)
000535      FEATURENAME(WorkloadIntCorr)
000536      STATE(ENABLED)
***** ***** Bottom of Data *****
```

PARMLIB(SMFPRMxx) (only required parameters shown)

```
EDIT          SYSx.PLXx.PARMLIB(SMFPRM00) - 01.71          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
***** ***** Top of Data *****
000500      ACTIVE /*ACTIVE SMF RECORDING*/
000510      WIC
000600      RECORDING(LOGSTREAM)
001510      INMEM(IFASMF.&SUBPLEX..WIC,RESSIZMAX(2G),TYPE(98))
***** ***** Bottom of Data *****
```

Getting WIC running(3/3)

T PROD=00

```
IFA100I IN PARMLIB MEMBER=IFAPRD00 ON LINE 532 208  
PRODUCTS WITH OWNER=IBM CORP NAME=z/OS  
FEATURE=WORKLOADINTCORR VERSION=*.*. ID=5650-ZOS  
HAVE BEEN ENABLED.
```

T SMF=00

```
T SMF=00  
IFA711I LOGSTREAM PARAMETERS ARE IN EFFECT  
IFA714I 11.35.18 SMF STATUS 554
```

LOGSTREAM NAME	BUFFERS	STATUS
A-IFASMF. .DEFAULT	550228	CONNECTED
A-IFASMF. .WIC	0	IN-MEMORY
A-IFASMF. .PERF	3216	CONNECTED
A-IFASMF. .CICSDB2	25448	CONNECTED
A-IFASMF. .SECURITY	2191	CONNECTED
A-IFASMF. .ZFS	12500	CONNECTED
A-IFASMF. .DPT	0	CONNECTED

```
IEE536I SMF VALUE 00 NOW IN EFFECT
```


DONE

D SMF,WIC

```
IFA714I 11.48.50 SMF STATUS 168
SMF WIC STATUS
SPECIFIED SMF PARAMETER: WIC
WorkloadIntCorr PRODUCT FEATURE: ENABLED
ST      K PG # AS  ROUTINE  VERSION  R A E LAST ROUTINE CALL TIME
----- - - - - -
00001          IEAHFXSV          Y      03/06/2022 11:48:45.706
01024 8 04 00006 DFHWICEX 00000002 Y Y N 03/06/2022 11:48:45.708
KEY:
ST      - SUBTYPE NUMBER
K        - BUFFERKEY VALUE
PG        - NUM4KPAGES VALUE
# AS      - NUMBER OF ADDRESS SPACES REGISTERED
ROUTINE   - EXIT ROUTINE NAME (CURRENT / PENDING)
VERSION   - EXIT ROUTINE VERSION (IN HEXADECIMAL)
R          - SMF PARAMETERS REQUEST SUBTYPE
A          - EXIT ROUTINE TO BE CALLED NEXT INTERVAL
E          - ERROR ADDING EXIT ROUTINE
LAST...   - TIME EXIT ROUTINE LAST CALLED
```

HIS reads 98s

- An in-memory SMF object does not write to a logstream
- You will need someone or something to read these records
- In the case of WIC it is HIS(Hardware Instrumentation Services)
- Some setup is needed to get HIS to read(and write out) the SMF 98 records

Setting up HIS

D HIS,WIC

```
HIS015I 09.33.43 DISPLAY HIS  
HIS WIC STATUS: INACTIVE  
INMEM= *NONE*  
WICPATH= *NONE*
```

F HIS,WIC,INMEM=IFASMF.PLXx.WIC,WICPATH='/tmp'

```
MODIFY HIS,WIC,INMEM=IFASMF. .PERF.INMEM,WICPATH='/tmp'  
HIS037I MODIFY HIS COMMAND FAILED. SPECIFIED INMEM INACCESSIBLE.  
IFAMCON SERVICE FAILED WITH RC=00000008, RSN=00000807.
```

You will get the above even if you are sure of the IN-MEMORY name. There will be no security messages but the way around this is by giving READ access to IFA.IFASMF.PLXx.WIC in the FACILITY class.

HIS is writing

D HIS,WIC

```
RESPONSE=ND
HIS01SI 16.55.32 DISPLAY HIS 601
HIS WIC STATUS: ACTIVE
INMEM= IFASMP.PU WIC
WICPATH= /tmp
```

```
Menu  Utilities  Compilers  Help
BROWSE  / /tmp/SYSWIC20220306.16.001.3
Command ----
2.....C.....%.....1.....OMVS .....
2.....C.....K.....FPQHWAM .....e;...
2.....C.....%.....1.....OMVS .....
2.....C.....K.....FPQHWAM .....e;...
2.....C.....%.....1.....OMVS .....
2.....C.....K.....FPQHWAM .....e;...
C..U.....NETD .....sA;.....0-X.....h.
C..U.....NETD .....sA;.....0-X.....h.
.....h.....S.....h.....X.....h.
.....IXGLOGR .....±A.....Se
.....IXGLOGR .....±A.....Se
.....h.....S.....h.....X.....h.
.....K.....IXGLOGR .....i
.....K.....IXGLOGR .....i
```

JOB Groups and the SCHEDULE JCL card

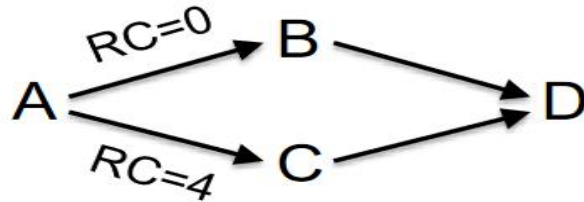
(Mike Shorkend)

Static Job Groups

z/OS 2.2

Overview – Simple Job Group Example

A four job example using dependencies and conditionals



//MULTI	JOBGROUP	JOBGROUP describes a group of dependent jobs
//A	GJOB	GJOB describes a job in a group
//B	GJOB	
//	AFTER NAME=A,	AFTER is positioned after a GJOB and describes its
//	WHEN= (RC=0)	dependency to another named job
//C	GJOB	
//	AFTER NAME=A,	
//	WHEN= (RC=4)	WHEN= describes a condition associated with the dependency
//D	GJOB	
//	AFTER NAME=B	
//	AFTER NAME=C	
//MULTI	ENDGROUP	ENDGROUP marks the end of the job group definition

Static JOB Groups

```
EDIT          MIKE.JCL.CNTL(JGONLY2) - 01.04          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
***** Top of Data *****
000100 //GROUP1 JOBGROUP
000300 //MIKEJOBA GJOB
000500 //MIKEJOBB GJOB
000700 // AFTER NAME=MIKEJOBA,
000900 // WHEN=(RC=0)
001100 //GROUP1 ENDGROUP
001200 // INCLUDE MEMBER=JGJOBS
***** Bottom of Data *****

  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help

EDIT          MIKE.JCL.CNTL(JHJOBS3) - 01.01          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
***** Top of Data *****
001300 //MIKEJOBA JOB (T00),'MIKE',MSGCLASS=X,NOTIFY=&SYSUID,TYPRUN=HOLD
001500 // SCHEDULE JOBGROUP=GROUP1
001800 //ALLOCATE EXEC PGM=IDCAMS,DYNAMNBR=200
001810 //SYSPRINT DD SYSOUT=*
001820 //SYSIN DD *
001840 SET MAXCC=8
001900 //MIKEJOBB JOB (T00),'MIKE',MSGCLASS=X,NOTIFY=&SYSUID
002100 // SCHEDULE JOBGROUP=GROUP1
002300 //S1 EXEC PGM=IEFBR14
***** Bottom of Data *****
```

The flow 1/2

- Submit Jobgroup; display SDSF/JG

```
Display Filter View Print Options Search Help
-----
SDSF JOB GROUP DISPLAY                               LINE 1-1 (1)
COMMAND INPUT ==>                                     SCROLL ==> CSR
PREFIX=*  DEST=(ALL)  OWNER=*  SYSNAME=
NP  JOBGROUP JobGrpID Owner   Status   Current-CC SAff Scheduling-Env  ErrStat
    GROUP1   G0002285 MIKE    PENDING
```

- Submit jobs; display SDSF/JG

```
-----
SDSF JOB GROUP DISPLAY                               LINE 1-1 (1)
COMMAND INPUT ==>                                     SCROLL ==> CSR
PREFIX=*  DEST=(ALL)  OWNER=*  SYSNAME=
NP  JOBGROUP JobGrpID Owner   Status   Current-CC SAff Scheduling-Env  Err
JP  GROUP1   G0002285 MIKE    ACTIVE
```


The flow 2/2

- Job Dependency panel

```
SDSF DEPENDENCY DISPLAY - GROUP GROUP1 (G0002285) LINE 1-1 (1)
COMMAND INPUT ==> S SCROLL ==> CSR
PREFIX=* DEST=(ALL) OWNER=* SYSNAME=
NP JOBNAME JobID Dependency DJobName DJobID Time When
   MIKEJOBA JOB02286 BEFORE MIKEJOBB JOB02287 RC=0000
```

- Release MIKEJOBA (ends with RC=8). Display group

```
12.54.32 G0002285 ---- SATURDAY, 19 MAR 2022 ----
12.54.32 G0002285 IRR010I USERID MIKE IS ASSIGNED TO THIS JOB.
S0W1 12.56.38 JOB02286 $HASP1300 MIKEJOBA registered to job group GROUP1
S0W1 12.56.38 JOB02286 $HASP1301 MIKEJOBA in job group GROUP1 queued for execution
S0W1 12.56.38 JOB02287 $HASP1300 MIKEJOBB registered to job group GROUP1
S0W1 13.01.09 JOB02286 $HASP373 MIKEJOBA STARTED - INIT 1 - CLASS A - SYS S0W1
S0W1 13.01.09 JOB02286 $HASP395 MIKEJOBA ENDED - RC=0008
S0W1 13.01.09 G0002285 $HASP1305 MIKEJOBB in job group GROUP1 is flushed
S0W1 13.01.09 G0002285 $HASP1304 job group GROUP1 is complete
```

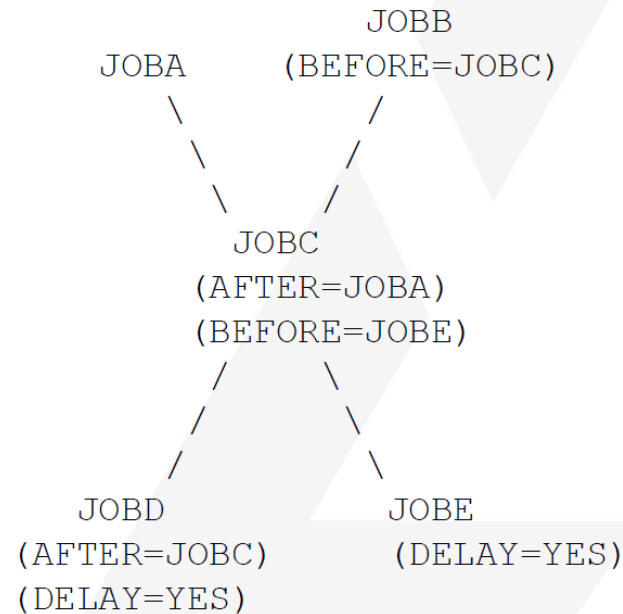
Dynamic Job Groups

z/OS 2.2; enhanced in 2.3

Dynamic job sequencing example

SHL
EDUCATE • NETWORK • INFLUENCE

Dynamic network:



```
//*-----  
//JOBE JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A  
// SCHEDULE DELAY=YES  
//STEP1 EXEC PGM=IEFBR14  
//*-----  
//JOB D JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A  
// SCHEDULE AFTER=JOBC,DELAY=YES  
//STEP1 EXEC PGM=IEFBR14  
//*-----  
//JOBC JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A  
// SCHEDULE BEFORE=JOBE,AFTER=JOBA  
//STEP1 EXEC PGM=IEFBR14  
//*-----  
//JOB B JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A  
// SCHEDULE BEFORE=JOBC  
//STEP1 EXEC PGM=IEFBR14  
//*-----  
//JOBA JOB TIME=NOLIMIT,REGION=0K,MSGCLASS=A,CLASS=A  
//STEP1 EXEC PGM=IEFBR14
```



```

EDIT          MIKE.JCL.CNTL(MIKE1) - 01.04          Columns 00001 00072
Command ==>  Scroll ==> CSR
***** Top of Data *****
000100 //MIKEJOB1      JOB NOTIFY=&SYSUID
000200 // EXEC PGM=IEFBR14
***** Bottom of Data *****

```

```

EDIT          MIKE.JCL.CNTL(MIKE2) - 01.06          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
***** Top of Data *****
000100 //MIKEJOB2 JOB NOTIFY=&SYSUID
000110 //          SCHEDULE BEFORE=MIKEJOB3,AFTER=MIKEJOB1,DELAY=YES
000200 // EXEC PGM=IEFBR14
***** Bottom of Data *****

```

```

EDIT          MIKE.JCL.CNTL(MIKE3) - 01.02          Columns 00001 00072
Command ==>           Scroll ==> CSR
***** Top of Data *****
000100 //MIKEJOB3      JOB NOTIFY=&SYSUID
000110 //      SCHEDULE DELAY=YES
000200 // EXEC PGM=IEFBR14
***** Bottom of Data *****

```

How does it flow?



- MIKEJOB2 has AFTER=MIKEJOB1. If there is a job named MIKEJOB1 after execution and on the spool, MIKEJOB2 will start immediately
- If there is no MIKEJOB1(after execution) on the spool, MIKEJOB2 will wait. On SDSF there is know indication why, however:

How does it flow?

```
- $DJ(2549),LONG
$HASP890 JOB(MIKEJOB2)
$HASP890 JOB(MIKEJOB2) STATUS=(AWAITING EXECUTION),CLASS=A,
$HASP890          PRIORITY=9,SYSAFF=(ANY),HOLD=(NONE),
$HASP890          CMDAUTH=(LOCAL),OFFS=(),SECLABEL=,
$HASP890          USERID=MIKE,SPOOL=(VOLUMES=(A3SYS1),
$HASP890          TGS=1,PERCENT=0.0040),ARM_ELEMENT=NO,
$HASP890          CARDS=3,DUBIOUS=NO,PRIVILEGE=NO,DJC=NO,
$HASP890          REBUILD=NO,SRVCLASS=BATMDM,SCHENV=,
$HASP890          SCHENV_AFF=(ANY),REQUIRES_ZOS=z/OS2.3,
$HASP890          CC=(),DELAY=(SCH_DEP),
$HASP890          CRTIME=(2022.082,18:28:07),
$HASP890          SCH_BEFORE=MIKEJOB3,SCH_AFTER=MIKEJOB1,
$HASP890          JOBGROUP=,
```

SCHEDULE keyword parameters

- BEFORE/AFTER/DELAY
- HOLDUNTIL= '+HH:MM:SS'
- HOLDUNTIL= 'HH:MM,MM/DD/YYYY'
- STARTBY= '+HH:MM:SS'
- STARTBY= 'HH:MM,MM/DD/YYYY'

Use the STARTBY parameter to specify the preferred date and time when the job should enter execution. JES attempts to position this job in the job queue in such a way that the job becomes ready to be selected for the execution at the specified time.

However, JES does **not guarantee** that the job starts executing at the specified time. The ability of the job to start executing depends on the system environment, system affinity, availability of initiators, availability of resources, and so on.

- WITH=JOBNAME – run on the same system as JOBNAME

SCHEDULE SDSF columns

```
SDSF INPUT QUEUE DISPLAY ALL CLASSES                LINE 1-1 (1)
COMMAND INPUT ==> █                                SCROLL ==> CSR
PREFIX=MIKEJOB*  DEST=(ALL)  OWNER=*  SYSNAME=
NP  JOBNAME  DelayRsn  HoldUntil  StartBy
    MIKEJOB2  HOLDUNTL  03/23/2022 20:50:37
```



If You Wanna Be Happy For The Rest of Your
Life

Make a PhD Grad Your ~~Wife~~ Sysprog
(James Lund)

We *Need* A New Sysprog



- Death of our Chief Sysprog
 - 45 years of experience lost
- Retirement of 2 other Sysprogs
- Our team of 5 quickly goes down to 2, then to ONE
 - Jack of all, Master of none
- Interview and hire a talented guy.
- Some other company steals them away a week later
- Other sysprog changes mind during the interview process
- Can't catch a break!

Here's a CRAZY idea!



What did we do?

We're a top tier university in Engineering, and we have a pool of top guys about to graduate and move out into industry or academia

Let's hire one of these guys!

Water, Water, Every Where... but ne'er a drop to drink...



- What did we need?
- Readily available - (close to) graduation
- Pre-trained in IT concepts
- Quick learner
- Love 24/7 work...
- NOT SO GOOD LUCK, UNTIL...

Shoot That Frozen Arrow...

- A PhD Grad student from Venezuela
- Already a PhD in Physics
- Working on a PhD in Aerospace
- Recently married
- Tired of school
- Not wanting to get into the rat race of research/publishing/etc.

- "So, do you know anything about mainframe?" "Never heard of it, but I'm willing to learn!"
- A long shot... what do we have to lose

- PAID OFF - He's a quick learner!

Forever Yours... Faithfully

- Hired
- He digs into books, RedPapers, websites
- Send him to IBM one week "basics" training
 - Comes back ready to change EVERYTHING!
- Asks LOTS of questions!
- Trial by fire - On Call duties
- Makes a few mistakes - nothing earth shattering

- 6 months later, takes charge of key function
- One year later, point person of key DR area
- 18 month later, swaps out automation system "as a training exercise"

...And They Lived Happily Ever After



Take Away of the Story

- Do you have bandwidth to train a new *insert title here*?
- Do you have time to wait for a newbie to spin up?
- Advertise at your local University for PhD candidates who:
 - Tired of the rush, rush, rush
 - Ready to do something practical vs academic
 - Gets married and has a baby on the way (well, maybe not *just* that one)



SEE YOU IN COLUMBUS!